[120] R. Tsoupidi, R. C. Lozano, and B. Baudry, "Constraint-based Diversification of JOP Gadgets," *J. Artif. Intell. Res.*, vol. 72, pp. 1471–1505, 2021.

[121] J. Falleri, F. Morandat, X. Blanc, M. Martinez, and M. Monperrus, "Fine-grained and Accurate Source Code Differencing," in *ACM/IEEE International Conference on Automated Software Engineering, ASE '14*, pp. 313–324, 2014.

[122] S. Banescu, C. Collberg, and A. Pretschner, "Predicting the Resilience of Obfuscated Code Against Symbolic Execution Attacks via Machine Learning," in *26th USENIX Security Symposium (USENIX Security 17)*, pp. 661–678, Aug. 2017.

[123] H. Bostani and V. Moonsamy, "EvadeDroid: A Practical Evasion Attack on Machine Learning for Black-box Android Malware Detection," *CoRR*, vol. abs/2110.03301, 2021.

[124] D. D. Yao, X. Shu, L. Cheng, and S. J. Stolfo, *Anomaly Detection as a Service: Challenges, Advances, and Opportunities*. Synthesis Lectures on Information Security, Privacy, and Trust, Morgan & Claypool Publishers, 2017.

[125] S. A. Hofmeyr, S. Forrest, and A. Somayaji, "Intrusion Detection Using Sequences of System Calls," *J. Comput. Secur.*, vol. 6, no. 3, pp. 151–180, 1998.

[126] Y. Fang, C. Huang, L. Liu, and M. Xue, "Research on Malicious JavaScript Detection Technology Based on LSTM," *IEEE Access*, vol. 6, pp. 59118–59125, 2018.

[127] E. Johnson, D. Thien, Y. Alhessi, S. Narayan, F. Brown, S. Lerner, T. McMullen, S. Savage, and D. Stefan, ", : SFI safety for native-compiled Wasm," *Network and Distributed Systems Security (NDSS) Symposium*, 2021.

[128] F. Cohen, "Computer Viruses," in *Proceedings of the 7th DoD/NBS Computer Security Conference 1984*, pp. 240–263, 1986.

[129] R. L. Castro, C. Schmitt, and G. D. Rodosek, "ARMED: How Automatic Malware Modifications Can Evade Static Detection?," in *2019 5th International Conference on Information Management (ICIM)*, pp. 20–27, 2019.

[130] R. L. Castro, C. Schmitt, and G. Dreo, "AIMED: Evolving Malware with Genetic Programming to Evade Detection," in *18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications / 13th IEEE International Conference On Big Data Science And Engineering, TrustCom/BigDataSE 2019, Rotorua, New Zealand, August 5-8, 2019*, pp. 240–247, IEEE, 2019.

[131] W. Wang, Y. Zheng, X. Xing, Y. Kwon, X. Zhang, and P. Eugster, "WebRanz: Web Page Randomization for Better Advertisement Delivery and Web-Bot Prevention," FSE 2016, p. 205–216, 2016.

[132] H. Aghakhani, F. Gritti, F. Mecca, M. Lindorfer, S. Ortolani, D. Balzarotti, G. Vigna, and C. Kruegel, "When Malware is Packin' Heat; Limits of Machine Learning Classifiers Based on Static Analysis Features," in *27th Annual Network and Distributed System Security Symposium, NDSS 2020, San Diego, California, USA, February 23-26, 2020*, The Internet Society, 2020.

[133] M. W. J. Chua and V. Balachandran, "Effectiveness of Android Obfuscation on Evading Anti-malware," in *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy, CODASPY*, pp. 143–145, 2018.

[134] P. Dasgupta and Z. Osman, "A Comparison of State-of-the-art Techniques for Generating Adversarial Malware Binaries," *CoRR*, vol. abs/2111.11487, 2021.

[135] G. Lu and S. K. Debray, "Weaknesses in Defenses against Web-borne Malware," in *Proceedings of Detection of Intrusions and Malware, and Vulnerability Assessment - 10th International Conference, DIMVA 2013*, vol. 7967, pp. 139–149, Springer, 2013.

[136] M. Payer, "Embracing the New Threat: Towards Automatically Self-diversifying Malware," in *Proceedings of The Symposium on Security for Asia Network*, pp. 1–5, 2014.

[137] N. Loose, F. Mächtle, C. Pott, V. Bezsmertnyi, and T. Eisenbarth, "Madvex: Instrumentation-based Adversarial Attacks on Machine Learning Malware Detection," in *Detection of Intrusions and Malware, and Vulnerability Assessment - 20th International Conference, DIMVA 2023*, vol. 13959 of *Lecture Notes in Computer Science*, pp. 69–88, 2023.

[138] A. V. Aho, R. Sethi, and J. D. Ullman, *Compilers: Principles, Techniques, and Tools*, ch. 1, pp. 28–31. 1986.

[139] R. Sasnauskas, Y. Chen, P. Collingbourne, J. Ketema, J. Taneja, and J. Regehr, "Souper: A Synthesizing Superoptimizer," *CoRR*, vol. abs/1711.04422, 2017.

[140] B. G. Ryder, "Constructing the Call Graph of a Program," *IEEE Transactions on Software Engineering*, no. 3, pp. 216–226, 1979.

[141] S. Narayan, C. Disselkoen, D. Moghimi, S. Cauligi, E. Johnson, Z. Gang, A. Vahldiek-Oberwagner, R. Sahita, H. Shacham, D. M. Tullsen, and D. Stefan, "Swivel: Hardening WebAssembly against Spectre," in *30th*